Princess Sumaya University for Technology
جامعة الأميرة سميّة للتكنولوجيا

The King Hussein School for Computing Sciences
Department of Computer Science

11103 - Structured Programming - Fall 2023

Midterm Exam

Full Name: Student ID:

Question	Points	Score
1	6	
2	8	
3	7	
4	9	
Total	30	

Circle your section:

○ Dr. Rawan Ghnemat (section 1)

○ Dr. Rawan Ghnemat (section 2)

○ Dr. Mohammad Abu Snober (section 3)

○ Dr. Rawan Ghnemat (section 4)

○ Dr. Mohammad Abu Snober (section 5)

Saturday 2/12/2023

Question 1 (6 points)

A. Implement a function named **`printLine`**. The function receives a character `ch` and two numbers: `n` and `s`. The function prints a line made of `s` spaces followed by the character `ch` repeated `n` times. The function returns without doing anything if `n` or `s` is negative.

Examples:

```
s = 3, n = 5, ch = H             output = ___HHHHH
s = 5, n = 5, ch = H             output = _____HHH
```

B. Implement a function named **`this_tank`**. The function receives an *odd* number `n` and prints a flipped triangle whose upper side is of length `n`. If `n` is even or negative, output an error message.

This function ***must*** call function `printLine`.

Example:

```
n = 5           output: VVVVV
                         VVV
                          V
```

C. Implement a function named **`this_this`**. The function receives an *odd* number `n` and prints a stack of flipped triangles that decrease in size. The upper triangle has an upper side of length `n` (see the example). If `n` is even or negative, output an error message.

This function ***must*** call function `this_tank`.

Example:

```
n = 7           output: VVVVVVV
                         VVVVV
                          VVV
                           V
                         VVVVV
                          VVV
                           V
                          VVV
                           V
                           V
```

Question 2 (8 points)

Write a C program that reads positive integers and stops as soon as a negative integer is entered. After reading every integer (except the first two), the program prints the maximum of the last 3 integers entered.

Example:

```
input  = 1 2 3 4 3 1 2 7
output = _ _ 3 4 4 4 3 7
```

In this example:

`3` is the max between `1 2 3`
`4` is the max between `2 3 4`
`4` is the max between `3 4 3`
`4` is the max between `4 3 1`
`3` is the max between `3 1 2`
`7` is the max between `1 2 7`

You are **not** allowed to use *arrays* in this question.

Question 3 (7 points)

Write a C program that defines an array of integers named `a[]` of size `1000` and fills it with random numbers between `100` and `900` (inclusive). After filling the array, the program must print out the numbers between `100` and `900` that are **not** in `a[]`.

Question 4 (9 points)

A. Implement a function named **`isEqual`**. The function receives two arrays (`a1[]` and `a2[]`), their sizes (`n` and `m`), and four indices: `i, j, k, h`. The function returns `1` if the elements `i⟶j` in `a1[]` are equal to the elements `k⟶h` in `a2[]`. The function returns `0` if they are not equal or if the arguments are invalid.

Examples:

```
a1 = [5, 2, 7, 3, 4, 5, 9]          a2 = [7, 3, 4, 5, 9, 1]
      0  1  2  3  4  5  6                 0  1  2  3  4  5
```

The function returns `1` for:

```
i=2, j=5,   k=0, h=3
i=4, j=5,   k=2, h=3
etc.
```

The function returns `0` for:

```
i=2,  j=5,   k=0, h=1
i=-2, j=5,   k=3, h=99
etc.
```

B. Implement a function named **isSubArray**. The function receives two arrays (a1[] and a2[]) and their sizes (n and m). The function returns 1 if a1[] is found inside a2[] and 0 otherwise.

Examples:	a1 = [5, 2, 7, 3]	a2 = [3, **5, 2, 7, 3**, 9, 1]	return 1
	a1 = [5, 2, 7, 3]	a2 = [5, 2, 0, 0, 7, 3, 9, 1]	return 0

This function ***must*** call function isEqual.

C. Implement a function named **canPartition** which receives an array, its size and a number w. The function checks if the array can be subdivided into consecutive equal arrays of size w each.

Examples:	a = [1, 2, 3, 1, 2, 3, 1, 2, 3]	size = 9, w = 3	return 1
	a = [1, 2, 1, 2, 1, 2, 1, 2]	size = 8, w = 2	return 1
	a = [9, 9, 9, 9, 9, 9, 9, 9, 9]	size = 9, w = 4	return 0
	a = [9, 9, 9, 9, 9, 9, 9, 9, 9]	size = 9, w = 1	return 1
	a = [1, 1, 1, 2, 2, 2, 1, 1, 1]	size = 9, w = 3	return 0

This function ***must*** call function isEqual.